

# A Tighter Analysis of Setcover Greedy Algorithm for Test Set

Peng Cui<sup>1</sup>

Key Laboratory of Data Engineering and Knowledge Engineering,  
Renmin University of China, Beijing 100872, China,  
cuiPeng@ruc.edu.cn

**Abstract.** Setcover greedy algorithm is a natural approximation algorithm for test set problem. This paper gives a precise and tighter analysis of approximation guarantee of this algorithm. The author improves the performance guarantee  $2 \ln n$  which derives from set cover problem to  $1.1354 \ln n$  by applying the potential function technique. In addition, the author gives a nontrivial lower bound  $1.0004471 \ln n$  of performance guarantee of this algorithm. This lower bound, together with the matching bound of information content heuristic, confirms the fact information content heuristic is slightly better than setcover greedy algorithm in worst case.

## 1 Introduction

The test set problem is NP-hard. The polynomial time approximation algorithms using in practice includes "greedy" heuristics implemented by set cover criterion or by information criterion[1]. Test set can not be approximated within  $(1-\varepsilon) \ln n$  for any  $\varepsilon > 0$  unless  $NP \subseteq DTIME(n^{\log \log n})$ [2,3]. Recently, the authors of [3] designed a new information type greedy algorithm, information content heuristic (ICH for short), and proved its performance guarantee  $\ln n + 1$ , which almost matches the inapproximability results.

The setcover greedy algorithm (SGA for short) is a natural approximation algorithm for test set. In practice, its average performance is virtually the same as information type greedy algorithms[1,4]. The performance ratio guarantee  $2 \ln n$  of SGA is obtained by transforming the test set problem as a set cover problem. The authors of [2] give the tight performance guarantee  $11/8$  of SGA on instances with the size of tests no greater than 2.

Oblivious rounding, a derandomization technique to obtain simple greedy algorithm for set cover problems by conditional probabilities was introduced in [5]. Young observed the number of elements uncovered is an "potential function" and the approximation algorithm only need to drive down the potential function at each step, thus he showed another proof of the well-known performance guarantee  $\ln n + 1$ .

In this paper, the author presents a tighter analysis of SGA. We uses the potential function technique of [5] to improve the performance guarantee  $2 \ln n$

which derives from set cover problem to  $1.1354 \ln n$ , and construct instances to give a nontrivial lower bound  $1.0004471 \ln n$  of the performance guarantee. The latter result confirms the fact ICH is slightly better than SGA in worst case. In this analysis, the author refers to the tight analysis of the greedy algorithm for set cover problem in [6].

In Section 2, the author show the two main theorems, and some definitions, notations and facts are given. In Section 3, the author analyzes differentiation distribution of item pairs and uses the potential function method to prove the improved performance guarantee. In Section 4, the author shows the nontrivial lower bound by constructing certain test set instances. Section 5 is some discussions.

## 2 Overview

The input of test set problem consists of  $S$ , a set of items (called universe), and  $\mathcal{T}$ , a collection of subsets (called tests) of  $S$ . A test  $T$  differentiates item pair  $\{i, j\}$  if  $|T \cap \{i, j\}| = 1$ .  $\mathcal{T}$  is a test set of  $S$ , that is, any item pair of  $S$  is differentiated by one test in  $\mathcal{T}$ . The objective is to find  $\mathcal{T}' \subseteq \mathcal{T}$  with minimum cardinality which is also a test set of  $S$ . We use  $\mathcal{T}^*$  to represent the optimal test set. Let  $n = |S|$ , and  $m^* = |\mathcal{T}^*|$ . In this paper, we assume  $m^* \geq 2$ .

In an instance of test set problem, there are  $\binom{n}{2}$  different item pairs. Let  $i, j$  be two different items, and  $S_1, S_2$  are two disjoint sets. If  $i, j \in S_1$ , we say  $\{i, j\}$  is an item pair inside of  $S_1$ , and if  $i \in S_1$  and  $i \in S_2$ , we say  $\{i, j\}$  is an item pair between  $S_1$  and  $S_2$ .

We use  $\{i, j\} \perp T$  to represent that  $T$  differentiates  $\{i, j\}$  and  $\{i, j\} \parallel T$  to represent that  $T$  does not differentiate  $\{i, j\}$ . We use  $\{i, j\} \perp \mathcal{T}$  to represent that at least one test in  $\mathcal{T}$  differentiates  $\{i, j\}$ ,  $\{i, j\} \parallel \mathcal{T}$  to represent that any test in  $\mathcal{T}$  does not differentiate  $a$ , and  $\perp (\{i, j\}, \mathcal{T})$  to represent the number of tests in  $\mathcal{T}$  that differentiate  $\{i, j\}$ .

**Fact 1.** For three different items  $i, j$  and  $k$ , if  $\{i, j\} \parallel \mathcal{T}$  and  $\{i, k\} \parallel \mathcal{T}$ , then  $\{j, k\} \parallel \mathcal{T}$ .

**Fact 2.** For three different items  $i, j$  and  $k$ , if  $\{i, j\} \perp \mathcal{T}$  and  $\{i, k\} \perp \mathcal{T}$ , then  $\{j, k\} \parallel \mathcal{T}$ .

Given  $\mathcal{T}' \subseteq \mathcal{T}$ , we define a binary relation  $\sim_{\mathcal{T}'}$  on  $S$ : for two item  $i, j$ ,  $i \sim_{\mathcal{T}'} j$  iff  $\{i, j\} \parallel \mathcal{T}'$ . By Fact 1,  $\sim_{\mathcal{T}'}$  is an equivalent relation. The equivalent classes containing  $i$  is denoted as  $[i]$ .

**Fact 3.** If  $\mathcal{T}$  is a minimal test set, then  $|\mathcal{T}| \leq n - 1$ .

**Fact 4.** If  $\mathcal{T}$  is a test set, then  $|\mathcal{T}| \geq \log_2 n$ .

Test set  $\mathcal{T}$  with  $|\mathcal{T}| = \log_2 n$  is called a compact test set. If  $\mathcal{T}$  is a compact test set, then  $|S| = 2^q, q \in \mathbb{Z}^+$ .

In set cover problem, we are given  $U$ , the universe, and  $\mathcal{C}$ , a collection of subsets of  $U$ .  $\mathcal{C}$  is a set cover of  $U$ , that is,  $\bigcup_{c \in \mathcal{C}} c = U$ . The objective is to find  $\mathcal{C}' \subseteq \mathcal{C}$  with minimum cardinality which is also a set cover of  $S$ .

The greedy algorithm for set cover runs like that. In each iteration, simply select a subset covering most uncovered elements, repeat this process until all

elements are covered, and return the set of selected subsets. Let  $N$  be the size of the universe, and  $M^*$  be the size of the optimal set cover. The greedy algorithm for set cover has performance guarantee  $\ln N - \ln \ln N + \Theta(1)$  by [6].

We give two lemmas about the greedy algorithm for set cover. Lemma 1 is a corollary of Lemma 2 in [6] and Lemma 2 is a corollary of Lemma 1 and Lemma 4 in [6].

**Lemma 1.** *The size of set cover returned by the greedy algorithm is at most  $M^*(\ln N - \ln M^* + 1)$ .*

**Lemma 2.** *Given  $N$  and  $M^*$ , there are instance of set cover problem such that the size of set cover returned by the greedy algorithm is at least  $(M^* - 1)(\ln N - \ln M^*)$ .*

Test set problem can be transformed to set cover problem in a natural way. Let  $(S, \mathcal{T})$  be an instance of test set, we construct an instance  $(U, \mathcal{C})$  of set cover, where  $U = \{\{i, j\} | i, j \in S, i \neq j\}$ , and  $\mathcal{C} = \{c(T) | T \in \mathcal{T}\}$ ,  $c(T) = \{\{i, j\} | i \in T, j \in T^-\}$ .

Clearly,  $\mathcal{T}'$  is a test set of  $S$  iff  $\mathcal{C}' = \{c(T) | T \in \mathcal{T}'\}$  is a set cover of  $U$ .

SGA can be described as:

```

Input:  $S, \mathcal{T}$ ;
Output: a test set of  $S$ ;
begin
     $\bar{\mathcal{T}} \leftarrow \emptyset$ ;
    while  $\#(\bar{\mathcal{T}}) > 0$  do
        select  $T$  in  $\mathcal{T} - \bar{\mathcal{T}}$  minimizing  $\#(\bar{\mathcal{T}} \cup \{T\})$ ;
         $\bar{\mathcal{T}} \leftarrow \bar{\mathcal{T}} \cup \{T\}$ ;
    endwhile
    return  $\bar{\mathcal{T}}$ ;
end

```

In SGA, we call  $\bar{\mathcal{T}}$  the partial test set. The differentiation measure of  $\bar{\mathcal{T}}$ ,  $\#(\bar{\mathcal{T}})$ , is defined as the number of item pairs not differentiated by  $\bar{\mathcal{T}}$ . The differentiation measure of  $T$  (related to  $\bar{\mathcal{T}}$ ) is defined as  $\#(T, \bar{\mathcal{T}}) = \#(\bar{\mathcal{T}}) - \#(\bar{\mathcal{T}} \cup \{T\})$ .

SGA is in fact isomorphic to the greedy algorithm for set cover under the natural transformation. Thus we immediately obtain the performance guarantee  $2 \ln n$  of SGA. This paper shows better performance guarantee and a nontrivial lower bound of performance guarantee. The two main theorems are:

**Theorem 1.** *The performance guarantee of SGA can be  $1.1354 \ln n$ .*

**Theorem 2.** *There are arbitrarily large instances of test set problem such that the performance ratio of SGA on these instances is at least  $1.0004471 \ln n$ .*

The harmonious number is defined as  $H_n := \sum_{i=1}^n 1/i$ .

Two inequalities are listed here for convenience of proof in Section 3.

**Fact 5.** *For any  $0 < x < 1$ ,  $(1 - x)^{1/x} < 1/e$ .*

Denote  $\phi(x) := \frac{1}{x}(\ln x - 1)$ .

**Fact 6.** *For any  $x > 1$ ,  $\phi(x) \leq 1/e^2 = 0.135 \dots$ .*

### 3 Improved Performance Guarantee

#### 3.1 Differentiation Distribution

In this subsection, the author analyzes the distribution of times for which item pairs are differentiated in instances of test set, especially the relationship between the differentiation distribution and the size of the optimal test set.

**Lemma 3.** *Given  $S_1, S_2 \subseteq S$ ,  $S_1 \cap S_2 = \emptyset$ , suppose  $\mathcal{T}$  is a test set of  $S_1$  and a test set of  $S_2$ , then at most  $\min(|S_1|, |S_2|)$  item pairs between  $S_1$  and  $S_2$  are not differentiated by any test in  $\mathcal{T}$ .*

*Proof.* Suppose  $|S_1| \leq |S_2|$ . We claim for any item  $i \in S_1$ , there exist at most one item  $j$  in  $S_2$  satisfying  $\{i, j\} \parallel \mathcal{T}$ . Otherwise there exist two different items  $j, k$  in  $S_2$  such that  $\{i, j\} \parallel \mathcal{T}$  and  $\{i, k\} \parallel \mathcal{T}$ , then by Fact 1,  $\{j, k\} \parallel \mathcal{T}$ , which contradicts  $\mathcal{T}$  is a test set of  $S_2$ .  $\square$

**Lemma 4.** *At most  $n \log_2 n$  item pairs are differentiated by exactly one test in  $\mathcal{T}^*$ .*

*Proof.* Let  $B$  be the set of item pairs that are differentiated by exactly one test in  $\mathcal{T}^*$ . We prove  $|B| \leq n \log_2 n$  by induction. When  $n = 1$ ,  $|B| = 0 = n \log_2 n$ . Suppose the proposition holds for any  $n \leq h - 1$ , we prove the proposition holds for  $n = h$ .

Select  $T \in \mathcal{T}^*$  such that  $T \neq \emptyset$  and  $T \neq S$ , then  $|T| \leq h - 1$ ,  $|S - T| \leq h - 1$ . Since  $\mathcal{T}^*$  is a test set of  $T$ , by induction hypothesis, at most  $|T| \log_2 |T|$  item pairs inside of  $T$  are differentiated by exactly one test in  $\mathcal{T}^*$ . Similarly, at most  $|S - T| \log_2 |S - T|$  item pairs inside of  $S - T$  are differentiated by exactly one test in  $\mathcal{T}^*$ .

By Lemma 3, at most  $\min(|T|, |S - T|)$  item pairs between  $T$  and  $S - T$  are not differentiated by any test in  $\mathcal{T}^* - \{T\}$ . Therefore at most  $\min(|T|, |S - T|)$  item pairs between  $T$  and  $S - T$  are differentiated by exactly one test in  $\mathcal{T}^*$ .

W.l.o.g, suppose  $|T| \leq |S - T|$ , then

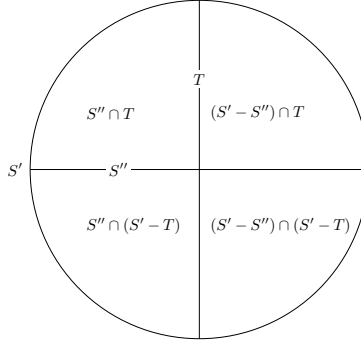
$$\begin{aligned} |B| &\leq |T| \log_2 |T| + |S - T| \log_2 |S - T| + T \\ &= |T| \log_2 (2|T|) + |S - T| \log_2 |S - T| \\ &\leq |T| \log_2 |S| + |S - T| \log_2 |S| \\ &= |S| \log_2 |S|. \end{aligned}$$

$\square$

**Lemma 5.** *Given  $S'' \subseteq S' \subseteq S$ , suppose  $\mathcal{T}$  is a test set of  $S''$  and a test set of  $S' - S''$ , then at most  $|S'| \log_2 |S'|$  item pairs between  $S''$  and  $S' - S''$  are differentiated by exactly one test in  $\mathcal{T}$ .*

*Proof.* Let  $B$  be the set of item pairs between  $S''$  and  $S' - S''$  which are differentiated by exactly one test in  $\mathcal{T}$ . We prove that  $|B| \leq |S'| \log_2 |S'|$  by induction. When  $|S'| = 1$  and  $|S'| = 2$ , the lemma holds clearly. Suppose the lemma holds for any  $|S'| \leq h - 1$ ,  $h \geq 3$ , we prove the lemma holds for  $|S'| = h$ .

Select  $T \in \mathcal{T}$  such that  $T \neq \emptyset$  and  $T \neq S'$ , then  $|T| \leq h - 1$ ,  $|S' - T| \leq h - 1$ . Since  $\mathcal{T} - \{T\}$  is a test set of  $S'' \cap T$  and a test set of  $(S' - S'') \cap T$ , by induction



**Fig. 1.** illustration of Lemma 5

hypothesis, at most  $|T| \log_2 |T|$  item pairs between  $S'' \cap T$  and  $(S' - S'') \cap T$  are differentiated by exactly one test in  $\mathcal{T}$ . Similarly, at most  $|S' - T| \log_2 |S' - T|$  item pairs between  $S'' \cap (S' - T)$  and  $(S' - S'') \cap (S' - T)$  are differentiated by exactly one test in  $\mathcal{T}$ .

Since  $\mathcal{T} - \{T\}$  is a test set of  $S'' \cap T$  and a test set of  $(S' - S'') \cap (S' - T)$ , by Lemma 3, at most  $\min(|S'' \cap T|, |(S' - S'') \cap (S' - T)|)$  item pairs between  $S'' \cap T$  and  $(S' - S'') \cap (S' - T)$  are not differentiated by any test in  $\mathcal{T} - \{T\}$ . Hence at most  $\min(|S'' \cap T|, |(S' - S'') \cap (S' - T)|)$  item pairs between  $S'' \cap T$  and  $(S' - S'') \cap (S' - T)$  are differentiated by exactly one test in  $\mathcal{T}$ . Similarly, at most  $\min(|(S' - S'') \cap T|, |S'' \cap (S' - T)|)$  item pairs between  $(S' - S'') \cap T$  and  $S'' \cap (S' - T)$  are differentiated by exactly one test in  $\mathcal{T}$ .

Clearly,

$$|T| \geq \min(|S'' \cap T|, |(S' - S'') \cap (S' - T)|) + \min(|(S' - S'') \cap T|, |S'' \cap (S' - T)|).$$

W.l.o.g, suppose  $|T| \leq |S' - T|$ , then

$$\begin{aligned} |B| &\leq |T| \log_2 |T| + |S' - T| \log_2 |S' - T| + |T| \\ &= |T| \log_2 (2|T|) + |S' - T| \log_2 |S' - T| \\ &\leq |T| \log_2 |S'| + |S' - T| \log_2 |S'| \\ &= |S'| \log_2 |S'|. \end{aligned}$$

□

**Lemma 6.** *At most  $n \log_2 nm^{*t-1}$  item pairs are differentiated by exactly  $t$  test in  $\mathcal{T}^*$ , where  $t \geq 2$ .*

*Proof.* Let  $B_t$  be the set of item pairs that are differentiated by exactly  $t$  test in  $\mathcal{T}^*$ . For any combination  $\pi$  of  $t - 1$  tests in  $\mathcal{T}^*$ , let  $B_\pi$  be the subset of  $B_t$  such that each item pair in  $B_\pi$  is differentiated by any test in  $\pi$ .

Let  $\sim_\pi$  be the equivalent relation induced by  $\pi$ . For any equivalent class  $[i]$ , there is exactly one equivalent class  $[j]$ , such that each item pair between  $[i]$  and  $[j]$  is differentiated by any test in  $\pi$  (Fact 2).

Since  $\mathcal{T}^* - \pi$  is a test set of  $[i]$  and a test set of  $[j]$ , by Lemma 5, at most  $(|[i] \cup [j]|) \log_2 |[i] \cup [j]|$  item pairs between  $[i]$  and  $[j]$  are differentiated by exactly one test in  $\mathcal{T}^* - \pi$ . In another word, at most  $(|[i] \cup [j]|) \log_2 |[i] \cup [j]|$  item pairs between  $[i]$  and  $[j]$  are differentiated by exactly  $t$  tests in  $\mathcal{T}^*$ . Hence

$$|B_\pi| = \sum_{[i],[j]} |[i] \cup [j]| \log_2 |[i] \cup [j]| \leq n \log_2 n.$$

Therefore,

$$|B_t| \leq \sum_{\pi} |B_\pi| \leq \binom{m^*}{t-1} n \log_2 n \leq n \log_2 n m^{*t-1}.$$

□

**Lemma 7.** *At most  $2n \log_2 n m^{*t-1}$  item pairs are differentiated by at most  $t$  test in  $\mathcal{T}^*$ , where  $t \geq 2$ .*

*Proof.* Let  $B$  be the set of item pairs that are differentiated by at most  $t$  test in  $\mathcal{T}^*$ , and  $B_t$  be the set of item pairs that are differentiated by exactly  $t$  test in  $\mathcal{T}^*$ . By Lemma 6,

$$\begin{aligned} |B| &= |B_1| + |B_2| + \cdots + |B_t| \\ &\leq n \log_2 n (1 + m^* + \cdots + m^{*t-1}) \\ &\leq 2n \log_2 n m^{*t-1}. \end{aligned}$$

gu □

### 3.2 Proof of Theorem 1

In this subsection, the author uses the potential function technique to derive improved performance guarantee of SGA for test set. Our proof is based on the trick to "balance" the potential function by appending a negative term to the differentiation measure.

Let  $I = \lceil \ln \frac{n-1}{4 \log_2 n} / \ln m^* \rceil$ , then  $2n \log_2 n m^{*I-1} < \binom{n}{2} \leq 2n \log_2 n m^{*I}$ . Let  $\#_0 = 1$ ,  $\#_1 = n \log_2 n$ ,  $\#_t = 2n \log_2 n m^{*t-1}$ ,  $2 \leq t \leq I$ , and  $\#_{I+1} = n(n-1)/2$ . Let  $k_t = \frac{m^*}{t} \ln \frac{t\#_t}{\#_{t-1}}$ ,  $2 \leq t \leq I+1$ .

Denote by  $p$  the probability distribution on tests in  $\mathcal{T}^*$  drawing one test uniformly from  $\mathcal{T}^*$ . For any  $T \in \mathcal{T}^*$ , the probability of drawing  $T$  is  $p(T) = \frac{1}{m^*}$ .

We divide a run of the algorithm into  $I+1$  phases, from Phase  $I+1$  to Phase 1. In Phase  $t$ ,  $I+1 \geq t \geq 1$ , the algorithm runs until  $\#(\bar{\mathcal{T}}) < \#_{t-1}$ . Let the set of selected tests in Phase  $t$  is  $\mathcal{T}_t$ , and the partial test set when Phase  $t$  stops is  $\bar{\mathcal{T}}_t$ ,  $1 \leq t \leq I+1$ . Then  $\bar{\mathcal{T}}_t = \cup_{1 \leq s \leq I+1} \mathcal{T}_s$ ,  $1 \leq t \leq I+1$ , and the returned test set is  $\mathcal{T}' = \cup_{1 \leq t \leq I+1} \mathcal{T}_t$ . Set  $\bar{\mathcal{T}}_{I+2} = \emptyset$ . If  $\mathcal{T}_t \neq \emptyset$ , let the last selected test in Phase  $t$  is  $T'_t$ .

In Phase  $t$ ,  $I+1 \geq t \geq 2$ , define the potential function as

$$f(\bar{\mathcal{T}}) = (\#(\bar{\mathcal{T}}) - \frac{t-1}{t} \#_{t-1}) (1 - \frac{t}{m^*})^{k_t - |\bar{\mathcal{T}} - \bar{\mathcal{T}}_{t+1}|}.$$

By the definition of  $\bar{\mathcal{T}}_{t+1}$  and Fact 5,

$$f(\bar{\mathcal{T}}_{t+1}) < (\#_t - \frac{t-1}{t}\#_{t-1})(1 - \frac{t}{m^*})^{k_t} < \frac{\#_{t-1}}{t}.$$

By the definition of  $f(\bar{\mathcal{T}})$  and the facts  $p(T) \geq 0$  and  $\sum_{T \in \mathcal{T}^*} p(T) = 1$ ,

$$\begin{aligned} & \min_{T \in \bar{\mathcal{T}}} f(\bar{\mathcal{T}} \cup \{T\}) \\ & \leq \min_{T \in \mathcal{T}^*} f(\bar{\mathcal{T}} \cup \{T\}) \\ & \leq \sum_{T \in \mathcal{T}^*} (p(T) f(\bar{\mathcal{T}} \cup \{T\})) \\ & = (\#(\bar{\mathcal{T}}) - \frac{t-1}{t}\#_{t-1} - \sum_{T \in \mathcal{T}^*} (p(T) \#(T, \bar{\mathcal{T}}))) (1 - \frac{t}{m^*})^{k_t - |\bar{\mathcal{T}} - \bar{\mathcal{T}}_{t+1}| - 1} \\ & = (\#(\bar{\mathcal{T}}) - \frac{t-1}{t}\#_{t-1} - \sum_{\{i,j\} \parallel \bar{\mathcal{T}}} \sum_{T \in \mathcal{T}^*: \{i,j\} \perp T} p(T)) (1 - \frac{t}{m^*})^{k_t - |\bar{\mathcal{T}} - \bar{\mathcal{T}}_{t+1}| - 1} \end{aligned}$$

and by Lemma 4 and Lemma 7,

$$\begin{aligned} & \sum_{\{i,j\} \parallel \bar{\mathcal{T}}} \sum_{T \in \mathcal{T}^*: \{i,j\} \perp T} p(T) \\ & \geq \sum_{\{i,j\} \parallel \bar{\mathcal{T}}: \perp(\{i,j\}, \mathcal{T}^*) \leq t-1} \frac{1}{m^*} + \sum_{\{i,j\} \parallel \bar{\mathcal{T}}: \perp(\{i,j\}, \mathcal{T}^*) \geq t} \frac{t}{m^*} \\ & = \sum_{\{i,j\} \parallel \bar{\mathcal{T}}} \frac{t}{m^*} - \sum_{\{i,j\} \parallel \bar{\mathcal{T}}: \perp(\{i,j\}, \mathcal{T}^*) \leq t-1} \frac{t-1}{m^*} \\ & \geq (\#(\bar{\mathcal{T}}) - \frac{t-1}{t}\#_{t-1}) \frac{t}{m^*}. \end{aligned}$$

Therefore,

$$\begin{aligned} & \min_{T \in \bar{\mathcal{T}}} f(\bar{\mathcal{T}} \cup \{T\}) \\ & \leq (\#(\bar{\mathcal{T}}) - \frac{t-1}{t}\#_{t-1}) (1 - \frac{t}{m^*})^{k_t - |\bar{\mathcal{T}} - \bar{\mathcal{T}}_{t+1}|} \\ & = f(\bar{\mathcal{T}}). \end{aligned}$$

During Phase  $t$ , the algorithm selects  $T$  in  $\mathcal{T}$  to minimize  $f(\bar{\mathcal{T}} \cup \{T\})$ . Therefore,  $f(\bar{\mathcal{T}}_t - \{T'_t\}) \leq f(\bar{\mathcal{T}}_{t+1}) < \frac{\#_{t-1}}{t}$ .

On the other hand,  $\#(\bar{\mathcal{T}}_t - \{T'_t\}) \geq \#_{t-1}$  by definition of Phase  $t$ . Hence

$$\begin{aligned} & f(\bar{\mathcal{T}}_t - \{T'_t\}) \\ & \geq (\#_{t-1} - \frac{t-1}{t}\#_{t-1}) (1 - \frac{t}{m^*})^{k_t - |\mathcal{T}_t - \{T'_t\}|} \\ & = \frac{\#_{t-1}}{t} (1 - \frac{t}{m^*})^{k_t - |\mathcal{T}_t - \{T'_t\}|}. \end{aligned}$$

Therefore,  $(1 - \frac{t}{m^*})^{k_t - |\mathcal{T}_t - \{T'_t\}|} < 1$ , thus  $|\mathcal{T}_t - \{T'_t\}| < k_t$ , and  $|\mathcal{T}_t| < k_t + 1$ .  
To sum up,

$$\begin{aligned} |\bar{\mathcal{T}}_2| &< \sum_{2 \leq t \leq I+1} k_t + I \\ &= m^* \left( \sum_{2 \leq t \leq I+1} \frac{1}{t} \ln \frac{\#_t}{\#_{t-1}} + \sum_{2 \leq t \leq I+1} \frac{\ln t}{t} \right) + I. \end{aligned}$$

When all Phase  $t$ ,  $I+1 \geq t \geq 2$ , end, consider the instance of set cover  $(U, \mathcal{C})$ , where  $U = \{a | a \parallel \bar{\mathcal{T}}_2\}$  and  $\mathcal{C} = \{c(T) | c(T) \cap U \neq \emptyset\}$ . Clearly,  $|U| < \#_1$ . Let  $M^*$  be the size of the optimal set cover of this instance. Then  $|M^*| \leq m^*$ .

Consider the following two cases: (a)  $|M^*| \leq m^*/2$ ; (b)  $|M^*| > m^*/2$ .

In case (a),

$$|\mathcal{T}_1| \leq M^*(\ln \#_1 + 1) \leq m^* \left( \frac{1}{2} + o(1) \right) \ln n,$$

and

$$\begin{aligned} |\bar{\mathcal{T}}_2| &\leq m^* \left( \sum_{2 \leq t \leq I+1} \frac{1}{2} \ln \frac{\#_t}{\#_{t-1}} + \frac{1}{2} \ln^2(I+2) \right) + I \\ &= m^* \left( \frac{1}{2} + o(1) \right) \ln n. \end{aligned}$$

Hence

$$|\mathcal{T}'| = |\bar{\mathcal{T}}_2| + |\mathcal{T}_1| = m^*(1 + o(1)) \ln n.$$

In case (b), by Lemma 1,

$$|\mathcal{T}_1| \leq M^*(\ln \#_1 - \ln M^* + 1) = m^*((1 + o(1)) \ln n - \ln m^*),$$

and

$$\begin{aligned} |\bar{\mathcal{T}}_2| &< m^*(H_{I+1} \ln m^* + \frac{1}{2} \ln 2 + \frac{1}{2} \ln^2(I+2)) + I \\ &= m^* \left( \ln \frac{\ln n}{\ln m^*} \ln m^* + o(1) \ln n \right). \end{aligned}$$

Hence

$$|\mathcal{T}'| = |\bar{\mathcal{T}}_2| + |\mathcal{T}_1| \leq m^* \left( 1 + \phi \left( \frac{\ln n}{\ln m^*} \right) + o(1) \right) \ln n.$$

By Fact 6,

$$|\mathcal{T}'| \leq m^*(1.13533 \cdots + o(1)) \ln n.$$



## 4 Lower Bound

In this section, we discuss on a variation of test set problem. Given disjoint sets  $S^1, \dots, S^r$  and  $\mathcal{T}$ , set of subsets of the universe  $S = S^1 \cup \dots \cup S^r$ , we seek  $\mathcal{T}' \subseteq \mathcal{T}$  with minimum cardinality which is a test set of any  $S^p$  for  $1 \leq p \leq r$ . Denote the instance by  $(S^p; \mathcal{T})$ .

In our construction, we could use the split trick similar to that used in [2] to split  $S^1, \dots, S^r$  by  $O(\log |S|)$  tests. The splitting overhead could be ignored, provided that the size of the optimal solution is  $\Omega(|S|^c)$  for some constant  $c$ .

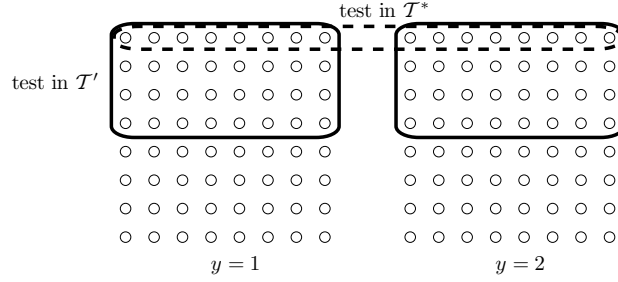
Let  $\hat{\mathcal{T}}$  is a compact test set of  $\{x | 1 \leq x \leq 2^q, x \in \mathbb{Z}^+\}$ . For example, we can let  $\hat{\mathcal{T}} = \{T_k | 1 \leq k \leq q\}$ , where  $T_k$  contains integer  $x$  between 1 and  $2^q$  such that the  $k$ -th bit of  $x$ 's binary representation is 1.

### 4.1 Level- $t$ Instances

Firstly, we give the level- $t$  atom instances. Let the instance be  $(S_t^y; \mathcal{T})$ . The universe  $S_t$  includes integral points in  $(t+1)$ -dimension Euclid space.

$S_t = \bigcup_y S_t^y$ ,  $S_t^y = \{(x_1, \dots, x_t, y) | 1 \leq x_i \leq 2^q, 1 \leq y \leq 2^{q-2}\}$ .  $\mathcal{T}_t = \mathcal{T}_t^* \cup \mathcal{T}_t'$ .  $\mathcal{T}_t^* = \mathcal{T}_{t,1}^* \cup \dots \cup \mathcal{T}_{t,t}^*$ .  $\mathcal{T}_{t,i}^* = \{T_{t,i,j}^* | 1 \leq j \leq 2^q\}$ .  $\mathcal{T}_t' = \mathcal{T}_{t,1}' \cup \dots \cup \mathcal{T}_{t,t}'$ ,  $\mathcal{T}_{t,i}' = \{T_{t,i,j,k}' | 1 \leq j \leq 2^{q-2}, 1 \leq k \leq q\}$ .

$T_{t,i,j}^*$  contains points in  $S_t^y$  with  $x_i = j$ .  $T_{t,i,j,k}'$  contains points in  $S_t^j$  with  $x_i$  in one of the  $q$  tests in  $\hat{\mathcal{T}}$ . We assign an order to tests  $T_{t,i,j,k}'$  in  $\mathcal{T}_t'$ , called natural order, as the lexical order of  $(i, j, k)$ .



**Fig. 2.** atom instance with  $q = 3$  and  $t = 2$

We claim SGA could return  $\mathcal{T}_t'$  according to their natural order on atom instances.

At the beginning of the algorithm, the differentiation measure of tests in  $\mathcal{T}_t'$  is  $2^{2qt-2}$ , and the differentiation measure of tests in  $\mathcal{T}_t^*$  is

$$2^{q(t-1)}(2^{qt} - 2^{q(t-1)})2^{q-2} = 2^{2qt-2}(1 - 2^{-q}).$$

The algorithm could first select tests in  $\mathcal{T}_{t,1}'$  according to their natural order. After that, the differentiation measure of tests in  $\mathcal{T}_t' - \mathcal{T}_{t,1}'$  decreases by a factor

2, and the differentiation measure of tests in  $\mathcal{T}_t^*$  decreases by a factor at least 2. Hence, the algorithm could subsequently select tests in  $\mathcal{T}'_{t,2}, \dots, \mathcal{T}'_{t,t}$  according to their natural order.

Secondly, we construct a series of level- $t$  instances  $(S_t^{y,z,w}; \mathcal{T}_t)$ ,  $1 \leq t \leq J$  based on the atom instances. Atom instances are "stretched" in  $w$  dimension and "cloned" in  $z$  dimension.

We select  $N$  and  $M^*$  such that  $M^* = J!2^q$  and  $N = J!2^{q(J+1)}$ , where  $q \in \mathbb{Z}^+$ ,  $J = 2^K - 1$  for some  $K \in \mathbb{Z}^+$ . The universe  $S_t$  includes  $N$  integral points in  $(t+3)$ -dimension Euclid space.

$S_t = \bigcup_{y,z,w} S_t^{y,z,w}$ ,  $S_t^{y,z,w} = \{(x_1, \dots, x_t, y, z, w) | 1 \leq x_i \leq 2^q\}$ ,  $1 \leq y \leq 2^{q-2}$ ,  $1 \leq z \leq \frac{J!}{t}$ ,  $1 \leq w \leq t2^{q(J-t)+2}$ .  $\mathcal{T}_t = \mathcal{T}_t^* \cup \mathcal{T}'_t$ .  $|\mathcal{T}_t^*| = M^*$ , and  $|\mathcal{T}'_t| = \frac{qM^*}{4}$ .  $\mathcal{T}_t^* = \mathcal{T}_{t,1}^* \cup \dots \cup \mathcal{T}_{t,t}^*$ .  $\mathcal{T}_{t,i}^* = \{T_{t,i,j,l}^* | 1 \leq j \leq 2^{q-2}, 1 \leq l \leq \frac{J!}{t}\}$ .  $\mathcal{T}'_t = \mathcal{T}'_{t,1} \cup \dots \cup \mathcal{T}'_{t,t}$ ,  $\mathcal{T}'_{t,i} = \{T'_{t,i,j,k,l} | 1 \leq j \leq 2^{q-2}, 1 \leq k \leq q, 1 \leq l \leq \frac{J!}{t}\}$ .

$\mathcal{T}_{t,i}^*$  contains points in  $S_t^{y,l,w}$  with  $x_i = j$ .  $\mathcal{T}'_{t,i,j,k,l}$  contains points in  $S_t^{j,l,w}$  with  $x_i$  in one of the  $q$  tests in  $\hat{\mathcal{T}}$ . We assign an order to tests  $T'_{t,i,j,k,l}$  in  $\mathcal{T}'_t$ , called natural order, as the lexical order of  $(i, j, k, l)$ .

We claim SGA could select all tests in  $\mathcal{T}'_{t,1}$  according to their natural order on  $(S_t^{y,z,w}; \mathcal{T}_t)$  in the beginning phase of the algorithm.

At the beginning of the algorithm, the differentiation measure of tests in  $\mathcal{T}'_{t,1}$  is  $\#_t^{begin} = 2^{q(t-1)}N$ , and the differentiation measure of tests in  $\mathcal{T}_t^*$  is  $2^{q(t-1)}(1 - 2^{-q})N$ .

The algorithm could select tests in  $\mathcal{T}'_{t,1}$  according to their natural order while the differentiation measure of selected test is kept equal to the differentiation measures of tests in  $\mathcal{T}'_{t,i}$  for  $2 \leq i \leq t$  and no less than the differentiation measure of any test in  $\mathcal{T}_t^*$ .

When the algorithm select the last test in  $\mathcal{T}'_{t,1}$ , its differentiation measure is  $\#_t^{end} = 2 \cdot 2^{q(t-2)}N$ .

## 4.2 Proof of Theorem 2

Let  $(U, \mathcal{C})$  be the instance in Lemma 2,  $U = \{e_1, \dots, e_N\}$ ,  $\mathcal{C}^*$  be the optimal set cover, and  $\mathcal{C}'$  be the set cover returned by the greedy algorithm. Construct instance of test set  $(S_0^p; \mathcal{T}_0)$ .  $S_0^p = \{e_p, f_p\}$ ,  $1 \leq p \leq N$ ,  $S_0 = \bigcup_{1 \leq p \leq N} S_0^p$ .  $\mathcal{T}_0 = \mathcal{C}^* \cup \mathcal{C}'$ .

On  $(S_0^p; \mathcal{T}_0)$ , the algorithm could select all the tests in  $\mathcal{T}_0'$ , the the differentiation measure of selected tests ranges from  $\#_0^{begin} = N/M^*$  to  $\#_0^{end} = 1$  by the proof of Lemma 1 in [6].

Consequently, we construct a series of level- $t$  instances  $(S_t^{y,z,w}; \mathcal{T}_t)$ ,  $1 \leq t \leq J$ , and combine them and  $(S_0^p; \mathcal{T}_0)$  into the complete instance. We intend to prove the performance ratio of SGA on this instance is at least  $(1 + \frac{1}{J+1}(\frac{H_J}{8 \ln 2} - 1) - o(1)) \ln n$ , for fixed  $J$ . When  $J = 511$ , this performance ratio is at least  $1.0004471 \ln n$ .

Let  $S = \bigcup_{t=0}^J S_t$ ,  $\mathcal{T} = \mathcal{T}^* \cup \mathcal{T}'$ . Then  $n = (J+1)N$ . We join tests in  $\mathcal{T}_t^*$  for  $0 \leq t \leq J$  one-by-one to obtain one test set  $\mathcal{T}^*$ . Suppose  $\mathcal{T}_t^* = \{T_{t,1}^*, \dots, T_{t,M^*}^*\}$ ,

$0 \leq t \leq J$ , then  $\mathcal{T}^* = \{T_{0,1}^* \cup \dots \cup T_{J,1}^*, \dots, T_{0,M^*}^* \cup \dots \cup T_{J,M^*}^*\}$ . Clearly,  $\mathcal{T}^*$  is an optimal test set, and  $m^* = M^*$ .

We modify tests in  $\mathcal{T}'_{t,1}$  by two operations: Enlargement and Merging. In the Enlargement operation, tests in  $\mathcal{T}'_{t,1}$  are enlarged by a factor 2. In the Merging operation, tests in  $\mathcal{T}'_{t,i}$  for  $i \geq 2$  are merged to tests in  $\mathcal{T}'_{s,1}$  for  $t-1 \geq s \geq 1$ . Let  $\mathcal{T}' = \mathcal{T}'_0 \cup \bigcup_{t=1}^J \mathcal{T}'_{t,1}$  after the two operations are performed.

**Enlargement.** Let  $\mathcal{T}'_{t,1} = \{T'_{t,1,j,k,l} | 1 \leq j \leq 2^{q-3}, 1 \leq k \leq q, 1 \leq l \leq \frac{J}{t}\}$ .  $T'_{t,1,j,k,l}$  contains points in  $S_t^{2j-1,l,w}$  and  $S_t^{2j,l,w}$  with  $x_1$  in one of the  $q$  tests in  $\mathcal{T}$ . As a result,  $|\mathcal{T}'_{t,1}| = \frac{qM^*}{8t}$ .

**Merging.** By the decreasing order of  $t$  for  $J \geq t \geq 2$ , merge tests in  $\mathcal{T}'_t - \mathcal{T}'_{t,1}$  by their natural order one-by-one to tests in  $\mathcal{T}'_{s,1}$  for  $t-1 \geq s \geq 1$  by the decreasing order of  $s$  (primarily) and their natural order in  $\mathcal{T}'_{s,1}$  until tests in  $\mathcal{T}'_t - \mathcal{T}'_{t,1}$  are exhausted.

For any  $J \geq t \geq 2$ , tests in  $\mathcal{T}'_{s,1}$  suffice in the Merging operation, provided that

$$|\mathcal{T}'_t - \mathcal{T}'_{t,1}| = (1 - \frac{1}{t}) \frac{qM^*}{4} \leq \frac{1}{2} H_{t-1} \frac{qM^*}{4} = |\bigcup_{s=1}^{t-1} \mathcal{T}'_{s,1}|.$$

To finish the proof, we analysis the behavior of SGA on the complete instance. Before the algorithm selects a test, let  $\#_t$  be the maximum differentiation measure of tests in  $\mathcal{T}'_{t,1}$  for  $J \geq t \geq 1$ , and  $\#^*$  to the maximum differentiation measure of tests in  $\mathcal{T}^*$ .

For  $1 \leq s \leq t$ , let  $\#_{t,s}$  be the number of item pairs inside of  $S_s^{y,z,w}$  contributing to  $\#_t$ ,  $\#_{t-1,s}$  be the number of item pairs inside of  $S_s^{y,z,w}$  contributing to  $\#_{t-1}$ ,  $\#_s^*$  be the number of item pairs inside of  $S_s^{y,z,w}$  contributing to  $\#^*$ , and  $\#_0^*$  be the number of item pairs inside of  $S_0^p$  contributing to  $\#^*$ . Then  $\#_t = \sum_{s=t}^J \#_{t,s}$ ,  $\#_{t-1} = \sum_{s=t-1}^J \#_{t-1,s}$ , and  $\#^* = \sum_{s=0}^J \#_s^*$ .

Since  $\#_{t,s} \geq \#_{t-1,s}$  for  $s > t$  and

$$\#_{t,t} = 2\#_{t-1,t} \geq \#_{t-1,t} + \#_t^{end} \geq \#_{t-1,t} + \#_{t-1}^{begin} \geq \#_{t-1,t} + \#_{t-1,t-1},$$

it follows that  $\#_t \geq \#_{t-1}$ . Hence  $\#_t \geq \#_s$ , for any  $1 \leq s \leq t$ .

Since  $\#_{t,s} \geq \#_s^*$  for  $s > t$  and

$$\#_{t,t} \geq \#_t^* + \#_t^{end} \geq \#_t^* + 2\#_{t-1}^{begin} \geq \#_t^* + \sum_{s=0}^{t-1} \#_s^{begin} \geq \#_t^* + \sum_{s=0}^{t-1} \#_s^*,$$

it follows that  $\#_t \geq \#^*$ .

We conclude the algorithm could select all tests in  $\mathcal{T}'_{t,1}$  in their natural order, for  $J \geq t \geq 1$ , and select all tests in  $\mathcal{T}'_0$ , finally return  $\mathcal{T}'$ .

In the condition  $J$  is fixed, the size of returned test set is

$$\begin{aligned} |\mathcal{T}'| &\geq (M^* - 1)(\ln N - \ln M^*) + \frac{qM^*}{8} H_J \\ &= m^* (1 + \frac{1}{J+1} (\frac{H_J}{8 \ln 2} - 1) - o(1)) \ln n. \end{aligned}$$

## 5 Discussion

The author notes this is the first time to distinguish precisely the worst case performance guarantees of two types of "greedy algorithms" implemented by set cover criterion and by information criterion. In fact, the author definitely shows the pattern of instances on which ICH performs better than SGA.

In a preceding paper[7], we proved the performance guarantee of SGA can be  $(1.5 + o(1)) \ln n$ , and the proof can be extended to weighted case, where each test is assigned a positive weight, and the objective is modified as to find a test set with minimum total weight.

In the minimum cost probe set problem[8] of bioinformatics, tests are replaced with partitions of items. The objective is to find a set of partitions with smallest cardinality to differentiate all item pairs. It is easily observed that the improved performance guarantee in this paper is still applicable to this generalized case.

**Acknowledgements.** The author would like to thank Tao Jiang and Tian Liu for their helpful comments.

## References

1. Moret B. M. E., Shpiro H. D.: On minimizing a set of tests. *SIAM Journal on Scientific and Statistical Computing* **6** (1985) 983–1003
2. De Bontridder K. M. J., Halldórsson B. V., Halldórsson M. M., Hurkens C. A. J., Lenstra J. K., Ravi R., Stougie L.: Approximation algorithm for the test cover problems. *Mathematical Programming-B* **98** (2003) 477–491
3. Berman P., DasGupta B., Kao M.: Tight approximability results for test set problems in bioinformatics, *Journal of Computer and System Sciences* **71** (2005) 145–162
4. DasGupta B., Konwar K., Mandoiu I., Shvartsman A.: Highly scalable algorithms for robust string barcoding, *International Journal of Bioinformatics Research and Applications* **1** (2005) 145–161
5. Young N. E.: Randomized rounding without solving the linear program, *Sixth ACM-SIAM Symposium on Discrete Algorithms (SODA 95)* 170–178
6. Slavík P.: A tight analysis of the greedy algorithm for set cover. *28th ACM Symposium on Theory of Computing (STOC 96)* 435–441
7. Cui P., Liu H.: Deep Approximation of Set Cover Greedy Algorithm for Test Set (in Chinese). *Journal of Software* **17** (2006) 1494–1500
8. Borneman J., Chrobak M., Vedova G. D., Figueora A., Jiang T.: Probe selection algorithms with applications in the analysis of microbial communities. *Bioinformatics* **17**(Suppl. 1) (2001) S39–S48